ZOMBIE TOWN

18 de mayo, 2020

ZOMBIE TOWN

Documento de investigación

Multimedia

ABSTRACT

En el siguiente documento, se presenta la investigación realizada para el desarrollo de un juego de video. Este juego fue desarrollado gracias al motor de juegos danés; Unity, con el propósito de poner en práctica los conocimientos se obtuvieron en la clase *programación de videojuegos*, además, de la búsqueda de conocimiento que sirvió para complementar este proyecto, ya que se utilizaron métodos de conocimiento que no se adquirieron directamente por esta clase, como el uso de algunas herramientas para desarrollo de gráficos que ayudaron en la elaboración de elementos se visualizaran el videojuego. El proyecto fue desarrollado con el lenguaje C#, lenguaje al cual se tuvo un mayor conocimiento gracias a la clase mencionada anteriormente. El videojuego que llamado “ZOMBIE TOWN” se encuentra publicado dentro de una ruta dentro de *GitHub*.

La ruta en la cual se encuentra el proyecto es la siguiente: https://github.com/joseatrejos/Unity-ZombieTown

ANTECEDENTES

Desde los principios del nuevo milenio, la rama de los videojuegos ha comenzado a tomar un importante posicionamiento dentro del mercado internacional, creciendo de manera inverosímil, ya que su uso no se encasilla únicamente para el entretenimiento personal, sino que ahora funciona como complemento para materiales didácticos sin importar a la rama laboral en la que se implemente, siendo *Unity*, uno de los principales motores para la creación de estos.

Un estudiante de la licenciatura en ingeniería en producción multimedia tiene los principios básicos para el procesamiento de imagen, audio y video, así como el desarrollo de softwares incluyendo videojuegos.

Para sacar provecho a los conocimientos de desarrollo de gráficos 3D, así como de programación, específicamente en el lenguaje *C#*. Se realizó un juego, creando escenarios, personajes, escenografía, animaciones, todo esto utilizando el modelado y texturizado que facilitó tres programas, tales como Maya familia de Autodesk, y Zbrush desarrollado por Pixologic.

Zombie town es un juego plataforma en el cual, el jugador es el encargado de controlar, al principio del juego, a un personaje principal que será el biólogo encargado de buscar preparar una cura, su objetivo es atravesar una ciudad llena de caos en donde a lo largo de recorrido tendrá que derrotar o esquivar a zombies y formar un equipo con otros sobrevivientes que encontrará en su camino.

DESARROLLO.

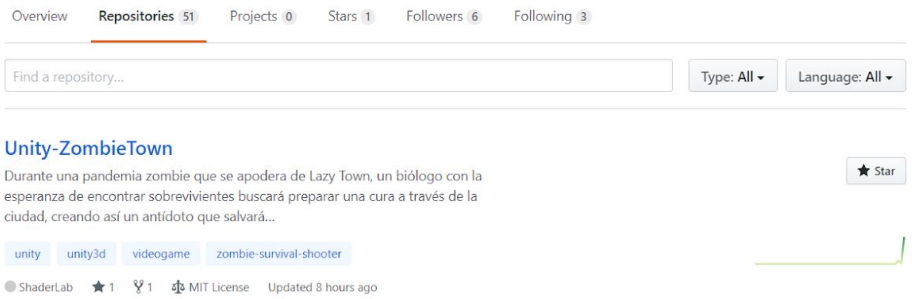
Para poder llevar a cabo la creación de este juego de video *(Zombie Town)* es necesario contar con un conjunto de tecnologías configuradas en los equipos de trabajo, ya que el proyecto, fue llevado a cabo por más de una persona, las cuales se encontraban trabajando a distancia, entre las cuales se encuentran las siguientes.

Git.

Esta herramienta es un sistema de comandos de consola que maneja el control de versiones de un proyecto, el cual está hosteado en un servidor remoto del cual podemos hacer clones locales en nuestro PC, modificarlos y trackear los cambios para publicarlos en este servidor remoto. Esta herramienta es utilizada en el proyecto con el fin de llevar el control del desarrollo de las versiones y así tener un mejor flujo de trabajo.

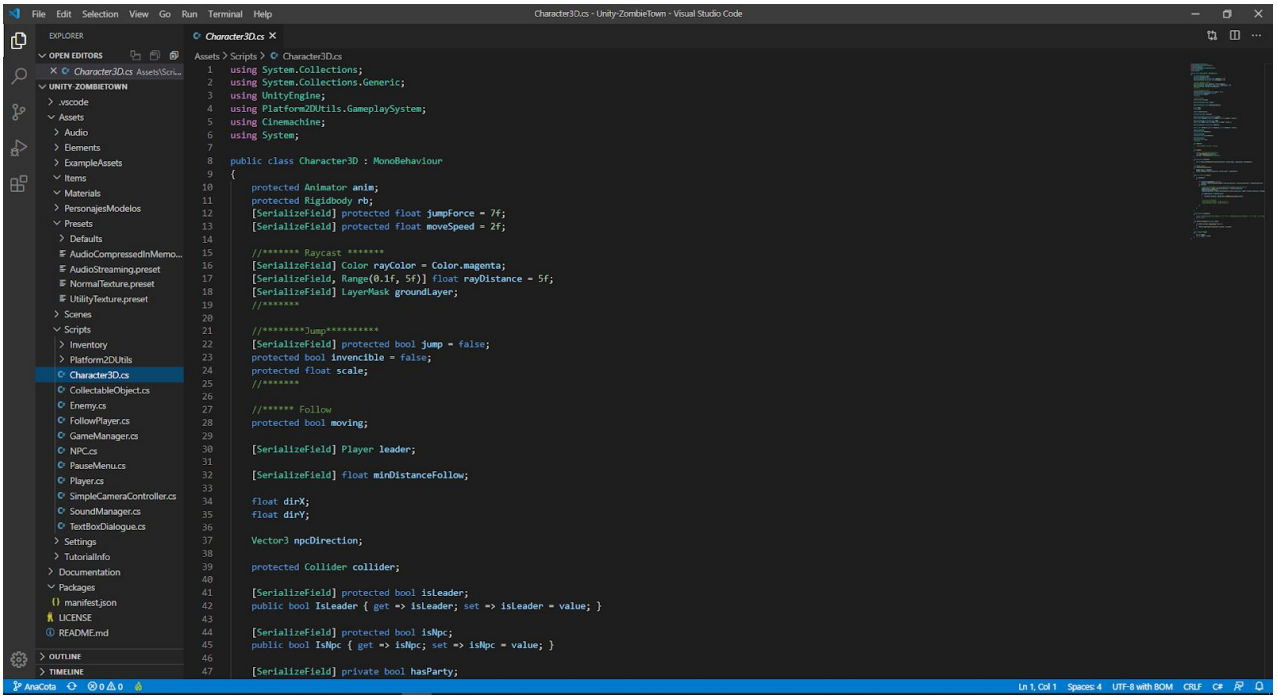
Para el host del proyecto, se maneja un repositorio de GitHub, ya que cuenta con un servicio de almacenamiento gratuito de gran capacidad.

José Alberto Trejo



Visual Studio Code (VS Code)

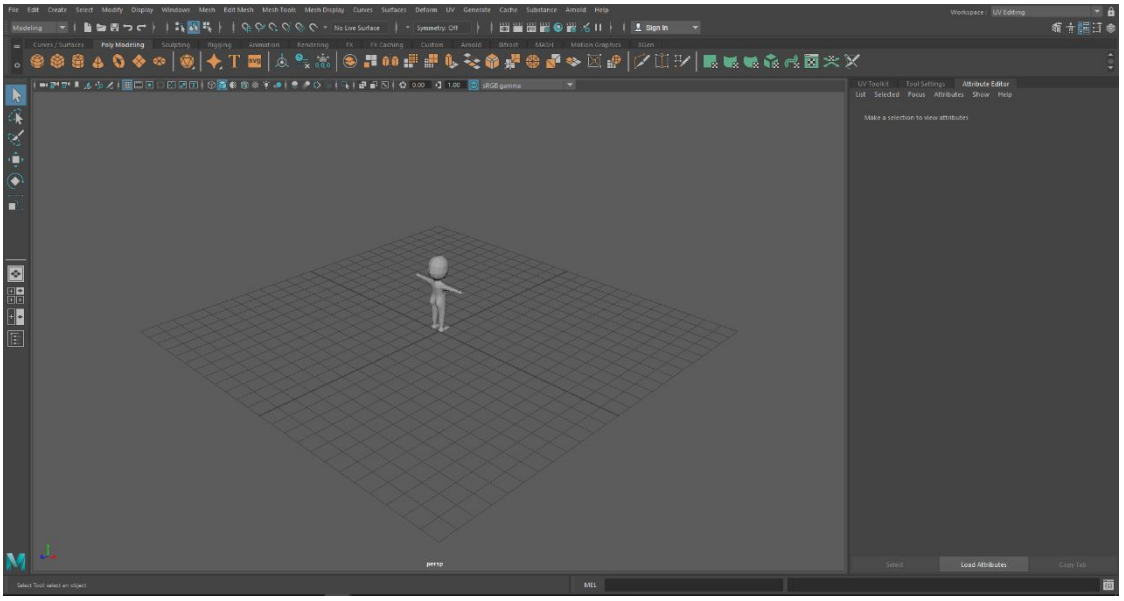
Este programa correspondiente a un editor de código con gran importancia estos últimos años, gracias a su facilidad de multitareas, así como también a su capacidad de soporte de compiladores externos, sin contar que tiene una línea de comandos powershell integrada con la que podremos trabajar fácilmente sin tener abierta la consola de forma externa.

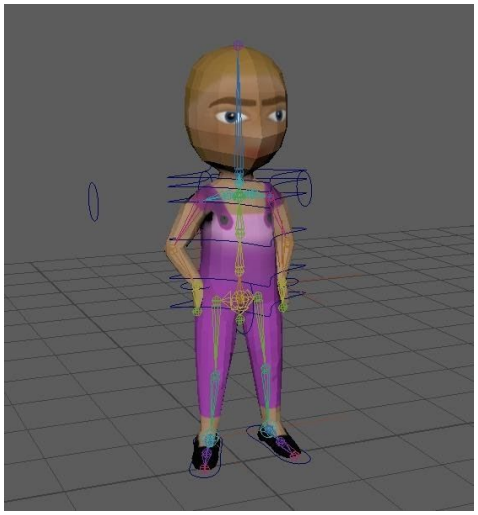


Fuera de la codificación, tres programas para desarrollo de gráficos fueron los necesarios para la creación de los *assets* utilizados dentro del videojuego.

Autodesk Maya

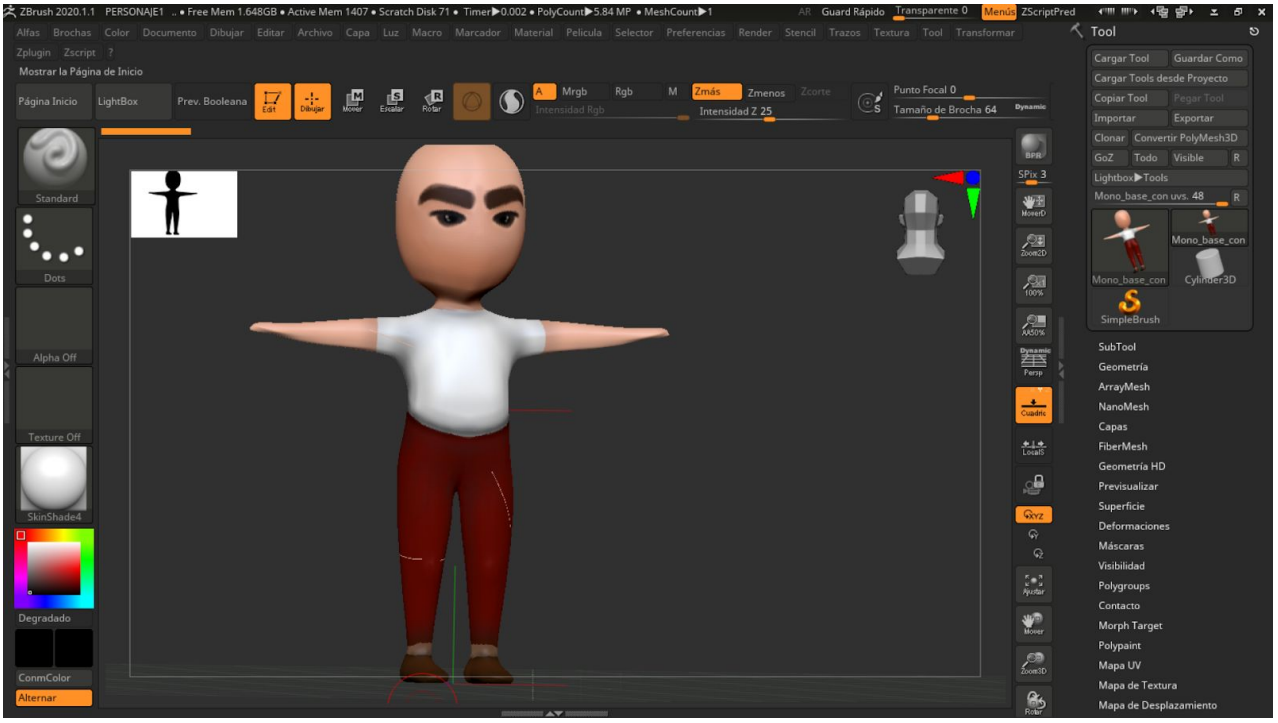
Este es un programa dedicado al desarrollo de gráficos 3D, efectos visuales y gráficos en movimiento. Este programa posee varias herramientas que pueden ser personalizadas para llevar a cabo cualquiera de los desarrollos que se mencionaron anteriormente. Dicho programa fue seleccionado para desarrollar la escenografía y personajes del videojuego, tanto en el modelado como en la animación de los gráficos.





Zbrush

Es un software especializado y dirigido principalmente a la escultura y pintura digital. El *polypaint* es una de las peculiaridades que tiene este programa, la cual permite pintar el *mapa UV* directamente sobre la superficie del modelo, asignando este color en el mapa. Este mapa se obtiene en el proceso de modelado desde Maya donde se proyecta una imagen 2D en la superficie de un modelo 3D. Después este mismo mapa que se exporto desde Zbrush se le asignó al modelo del personaje en Maya para comenzar con las animaciones mostradas anteriormente.



Para el funcionamiento de cada movimiento, fue necesario el uso de librerías complementarias para lograr un mejor flujo de juego.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Platform2DUtils.GameplaySystem;
using UnityEngine.AI;
```

```
using UnityEngine.UI;
using UnityEngine.SceneManagement;
```

Librerías utilizadas.

- De la siguiente librería tomamos el Gameplay System, más sin embargo se tuvo que modificar para adaptarla al Zombie Town 3D

```
public static void CheckDistance(Transform enemyTransform, Transform target, float chaseRadius, float moveSpeed)
{
    if(target)
    {
        if (Vector3.Distance(target.position, enemyTransform.position) <= chaseRadius)
        {
            enemyTransform.position = Vector3.MoveTowards(enemyTransform.position, target.position, moveSpeed * Time.deltaTime);
        }
    }
}

1 reference
public static void Movement3D(Transform transform, float moveSpeed)
{
    transform.Translate(Axis.normalized.magnitude * Vector3.forward * moveSpeed * Time.deltaTime);

}

5 references
public static  Vector3 Axis3D
{
    get => new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
}
```

- Script de movimiento de la party:

  Se utilizó la versión del anterior party pero se modificó de acuerdo a lo que se necesito para el juego.

```
public void SwapLeader()
{
    if (currentParty.Count > 1 && canChange)
    {
        Player currentLeader = currentParty[0];
        currentLeader.IsLeader = false;
        currentLeader.IsNpc = true;
        currentLeader.HasParty = true;
        currentLeader.Target = currentParty[currentParty.Count - 1];
        currentLeader.navMeshAgent.enabled = true;
        currentLeader.GetComponent<Collider>().isTrigger = true;
        currentParty.RemoveAt(0);
        currentLeader.gameObject.tag = "NPC";
        currentParty.Add(currentLeader);
        currentParty[0].gameObject.tag = "Player";
        currentParty[0].IsLeader = true;
        currentParty[0].IsNpc = false;
        currentParty[0].Target = null;
        currentParty[0].HasParty = true;
        canChange = false;
        currentParty[0].navMeshAgent.enabled = false;
        currentParty[0].GetComponent<Collider>().isTrigger = false;
        GameManager.instance.party.currentParty[0].ScaleLife();
    }
}
```

- Script Sound Manager

```
public void PlayAudio(AudioClip audioClip)
{
    if (!CompareAudio(audioClip))
    {
        audioSource.clip = audioClip;
        audioSource.Play();
    }
}

2 references
public void PlayBGM()
{
    PlayAudio(backgroundMusic);
}

1 reference
public void PlayChaseMusic()
{
    PlayAudio(chaseMusic);
}

1 reference
public void PlayCombatMusic()
{
    PlayAudio(combatMusic);
}

1 reference
public void WeaponDrawn()
{
    PlayAudio(weaponDrawn);
}

1 reference
public bool CompareAudio(AudioClip newAudioClip)
{
    return newAudioClip == audioSource.clip;
}
```

CONCLUSIONES

Con el tiempo los videojuegos se han posicionado muy alto ya que no se consideran solamente para fines de entretenimiento, sino para fines de aprendizaje también. Estudios nos dicen que los juegos de estrategia están relacionados con la mejora de los procesos ejecutivos de control de la mente. Zombie Town no es una excepción, ya que es indispensable la estrategia para poder terminar el juego agilizando la mente. Por el lado del desarrollo, el uso de la lectura complementa el conocimiento requerido para la creación del juego plataforma.

El resultado final del videojuego tiene al equipo muy satisfecho, ya que gracias a la organización temprana y con el tiempo disponible para el desarrollo se pudo completar satisfactoriamente el alcance de este proyecto.

REFERENCIAS.

1. Brackeys. (2017, marzo 19). GAME OVER - How to make a Video Game in Unity (E08) [Archivo de vídeo]. YouTube. Recuperado de https://www.youtube.com/watch?v=VbZ9_C4-Qbo
2. Change value of a projectile that is instantiated in Awake and pooled - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/1567259/change-value-of-a-projectile-that-is-instantiated.html
3. Disable collisions between NavMeshAgents - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/711588/disable-collisions-between-navmeshagents.html
4. [Help!] Getting movement direction of an enemy? (s. f.). Recuperado 15 de mayo de 2020, de https://forum.unity.com/threads/help-getting-movement-direction-of-an-enemy.418805/
5. How do I stop a rigidbody from rotating after colliding with another object? - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/808879/how-do-i-stop-a-rigidbody-from-rotating-after-coll.html
6. How to determine the direction an object is moving relative to itself? - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/689999/how-to-determine-the-direction-an-object-is-moving.html
7. How to use Collision.GetContacts - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/1612678/how-to-use-collisiongetcontacts.html
8. Push Object in Opposite Direction of Collision - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/1100879/push-object-in-opposite-direction-of-collision.html
9. SOLVED. How can I use one parent object animator to control multiple child sprite animations? - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/1437649/how-can-i-use-one-parent-object-animator-to-contro.html
10. Unity. (s. f.-a). Image. Recuperado 15 de mayo de 2020, de https://docs.unity3d.com/es/2018.4/ScriptReference/UI.Image.html
11. Unity. (s. f.-b). Utilizando Animation Events (Eventos de Animación). Recuperado 15 de mayo de 2020, de https://docs.unity3d.com/es/530/Manual/animeditor-AnimationEvents.html
12. What's a way to implement a flexible buff/debuff system? (2012, junio 1). Recuperado 15 de mayo de 2020, de https://gamedev.stackexchange.com/questions/29982/whats-a-way-to-implement-

a-flexible-buff-debuff-system

13. When a gameobject is destroyed a chunk of the navmesh gets destroyed - Unity Answers. (s. f.). Recuperado 15 de mayo de 2020, de https://answers.unity.com/questions/1381107/when-a-gameobject-is-destroyed-a-chunk-of-the-navm.html
14. Lengyel, Eric (2011). Principios Generales Y Matemáticas Aplicadas. Cengage Learning.
15. Lengyel, Eric (2013). Matemáticas Para Videojuegos En 3D. Cengage Learning. Goldstone, Will (s.f). Unity Game Development Essential. Packt Publishing.
16. Norton, Terry (2013). Learning C# by Developong Games with Unity 3D Beginner´s Guida. Packt Publishing.
17. Unity Documentation.. (2019). JSON Serialization. 2019, de Unity Sitio web: https://docs.unity3d.com/Manual/JSONSerialization.html
18. Johnson, R. E. (1987). Model/View/Controller. Department of C.S.
19. Jesús Francisco Caro Cota . (2019). diseño de una api rest publica informativa utilizando tecnología java spring boot y su consumo en un sitio web con javascript ecmascript 6 . en diseño de una api rest publica informativa utilizando tecnología java spring boot y su consumo en un sitio web con javascript ecmascript 6 (21). universidad da vinci: universidad da vinci.
20. https://github.com/Unity-Technologies/2d-extras#master
21. Mike Geig. (2014). Unity Game Development in 24 Hours. 2013, de Pearson Education, Inc Sitio web: http://docshare02.docshare.tips/files/28607/286070481.pdf
22. Unity.. (2013). Solution: Unity for mobile games . 2013, de Unity Sitio web: https://unity3d.com/files/solutions/unityformobile/Unity_For_Mobile_Guide.pdf
23. Alan Thorn. (2010). Learn Unity for 2D Game Development. 2010, de Apress Sitio web: https://www.academia.edu/28371929/Learn_Unity_for_2D_Game_Development_-_TIA
24. Alan Thorn. (2010). Learn Unity for 2D Game Development. 2010, de Apress
25. Simon Jackson. (2018). Unity 3D UI Essentials. 2018, de PacktPublishing Sitio web: https://mydevbook.info/topics/game-development/unity-game-development/
26. John P. Doran . (2019). Unity 3.x Scripting. 2019, de PacktPublishing Sitio web: https://mydevbook.info/topics/game-development/unity-game-development/
27. John P. Doran, Alan Zucconi. (2018). Unity 2018 Shaders and Effects Cookbook. Third Edition. 2018, de PacktPublishing Sitio web: https://mydevbook.info/topics/game-development/unity-game-development/
28. Pablo Farias Navarro. (2018). Learn Unity by Creating a 3D Multi-Level Platformer Game. 2018, de Zenva Sitio web: https://gamedevacademy.org/wp-content/uploads/2018/04/Learn-Unity-by-Creating-a-3D-Multi-Level-Platformer-Game.pdf
29. Matt Smith, Chico Queiroz. (2019). Unity 5.x Cookbook. 2019, de

PacktPublishing Sitio web: https://mydevbook.info/topics/gamedevelopment/unity- game-development/

30. Kyle D'Aous. (2019). Unity Game Development Scripting. 2019, de PacktPublishing Sitio web: https://mydevbook.info/topics/gamedevelopment/unity- game-development/
31. https://docs.unity3d.com/Manual/JSONSerialization.html
32. https://docs.unity3d.com/es/2018.4/Manual/Coroutines.html
33. Jate Wittayabundit. (2019). Unity 4 Game Development Hotshot. 2019, de PacktPublishing Sitio web: https://mydevbook.info/topics/game-development/unity- game-development/
34. https://answers.unity.com/questions/1427984/how-to-make-object-move-from-one-random-point-to-a.html
35. Jorge Palacios. (2019). Unity 5.x Game AI Programming Cookbook. 2019, de PacktPublishing Sitio web: https://mydevbook.info/topics/game-development/unity-game-development/page/2/
36. https://docs.unity3d.com/ScriptReference/Vector2.Lerp.html
37. https://docs.unity3d.com/Manual/JSONSerialization.html
38. Pixologic, Inc. (2020). ZBrush at a Glance. Recuperado de http://pixologic.com/features/about-zbrush.php
39. Pixologic, Inc. (2020b). ZBrush at a Glance. Recuperado de http://pixologic.com/features/about-zbrush.php
40. Autodesk. (2018, agosto 21). UVs | Maya 2018 | Autodesk Knowledge Network. Recuperado de https://knowledge.autodesk.com/support/maya/learn-explore/caas/CloudHelp/cloudhelp/2018/ENU/Maya-Modeling/files/GUID-FDCD0C68-2496-4405-A785-3AA93E9A3B25-htm.html
41. Painting Your Model | ZBrush Docs. (2020). Recuperado de http://docs.pixologic.com/user-guide/3d-modeling/painting-your-model/
42. GÓMEZ, P. M. (2018, abril 11). Paint onto a 3D mesh with ZBrushCore's Polypaint tool. Recuperado de https://www.creativebloq.com/how-to/paint-onto-a-3d-mesh-with-zbrushcores-polypaint-tool
43. Brackeys. (2018, February 11). OBJECT POOLING in Unity [Video file]. YouTube. Retrieved from https://www.youtube.com/watch?v=tdSmKaJvCoA
44. Blackthornprod. (2018, January 13). HOW TO EXPORT ANIMATIONS FROM MAYA TO UNITY - TUTORIAL [Video file]. YouTube. Retrieved from https://www.youtube.com/watch?v=i8cmDK_O-1g&t=
45. Can't add script component because the script class cannot be found? (2018, August 6). Retrieved May 16, 2020, from https://stackoverflow.com/questions/51713497/cant-add-script-component-becau

se-the-script-class-cannot-be-found

46. How to unstage large number of files without deleting the content. (2011, August 18). Retrieved May 16, 2020, from https://stackoverflow.com/questions/7103631/how-to-unstage-large-number-of-files-without-deleting-the-content

47. How to use Collision.GetContacts - Unity Answers. (n.d.). Retrieved May 16, 2020, from https://answers.unity.com/questions/1612678/how-to-use-collisiongetcontacts.html

48. Jason Weimann. (2018, November 15). Object Pooling (in depth) - Game Programming Patterns in Unity & C# [Video file]. YouTube. Retrieved from https://www.youtube.com/watch?v=uxm4a0QnQ9E&start=

49. Learn Everything Fast. (2017, March 23). Improving Game Performance with Object Pooling in Unity3d [Video file]. YouTube. Retrieved from https://www.youtube.com/watch?v=2TQzbC136jI

50. Single-Dimensional Arrays - C# Programming Guide. (2015, July 20). Retrieved May 16, 2020, from https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/single-dimensional-arrays

51. Unity. (n.d.). Animator. Retrieved May 16, 2020, from https://docs.unity3d.com/ScriptReference/Animator.html